Homework III, Advanced algorithms, Spring 1995

This homework is due Tuesday June 6 at 9 am.

1. In a lecture I outlined how to use the Fourier transform over the complex numbers to do multiplication of n bit numbers. In particular I said that one should consider the numbers to be multiplied as vectors of length $n/\log n$ where each number is an integer which is at most n. One should then work over the complex numbers with $O(\log n)$ bits of accuracy. Work out the details of this algorithm.

2. Let us cast the problem of integer multiplication in terms of general bilinear problems. We have inputs bits $(a_i)_{i=0}^{n-1}$ and $(b_i)_{i=0}^{n-1}$. If we insist on the standard output, then we do not have a bilinear problem because of the carries. Suppose instead we ask to compute

$$c_k = \sum_{i+j=k} a_i b_j$$

   for $k = 0, 1, \ldots 2n-2$. This is a bilinear problem and is given by a tensor. Show that the rank of the given tensor is $O(n)$. Why does this not give a linear time algorithm for integer multiplication?

3. Show that computing complex multiplication requires 3 real multiplications. In other words show that the rank of the corresponding tensor is exactly 3.

4. How many comparisons are needed to find the second largest element in a set of n elements? I want the answer on the form $f(n) + O(1)$, i.e. the complexity should be found up to an additive constant.

5. When presenting the simple median algorithm in class it was suggested that we should divide the inputs into larger groups and then take the median of those. Evaluate the alternative of dividing the inputs into groups of 7.

6. Consider the heuristic of nearest neighbor for TSP. Produce for arbitrary n an instance which obeys the triangle inequality such that the produced tour is much longer than the optimal tour. If the ratio to the shortest tour is a constant greater than 1, you get 1 p, if it grows to infinity with n you get 2 p and if it grow like $c \log n$ you get 3 p.